

Самостоятельная работа 1

Выполнил: Славный Даниил Михайлович

Группа: 1.2

### **Задание 3.2: Анализ БД на избыточность**

Избыточность данных в базе данных происходит, когда одна и та же информация хранится в нескольких местах. Хотя это может ускорить некоторые операции чтения, избыточность может привести к проблемам с обновлением, удалением и вставкой данных.

#### **Пример неверного логического проектирования:**

Предположим, у нас есть таблица "Заказы", которая содержит следующие поля: OrderID, CustomerName, CustomerAddress, ItemOrdered. В этом случае, если один и тот же клиент сделает несколько заказов, его имя и адрес будут повторяться в каждой записи. Это приводит к избыточности данных.

#### **Алгоритм решения:**

**Нормализация:** Это процесс разделения таблиц на меньшие, более управляемые таблицы без избыточности. В нашем примере мы можем разделить таблицу "Заказы" на две таблицы: "Клиенты" (с полями CustomerID, CustomerName, CustomerAddress) и "Заказы" (с полями OrderID, CustomerID, ItemOrdered). Теперь вместо того, чтобы повторять имя и адрес клиента для каждого заказа, мы просто ссылаемся на CustomerID.

**Обновление всех дубликатов:** При внесении изменений в информацию, необходимо обновить ее во всех местах, где она хранится. Это может быть сложно реализовать и привести к ошибкам.

**Удаление дубликатов:** В случае обнаружения дубликатов, они должны быть удалены. Это должно быть выполнено с осторожностью, чтобы не удалить важные данные.

**Проверка целостности данных:** Это важный этап, который обеспечивает, что все данные после преобразования остаются целыми и согласованными.

Нормализация уменьшает избыточность, но может замедлить некоторые операции запросов. Важно найти баланс между нормализацией и производительностью